PERSONATA

CS3216 – Assignment 1 : Group 05 Aspirations

Aspiration 1:

For our Facebook assignment 1, our group decided to build a standalone application. We chose to implement a standalone application over a canvas application is because standalone web application has more freedom of developments like user interface compared to a canvas app, of which size, space, theme as well as styles are all limited by the iframe.

Aspiration 2:

The name that we assigned to our application is Personata. This name is a play on the words 'personal data' as that is the kind of information that our application focuses on.

Aspiration 3:

Application icon was designed for the mid-assignment submission.

Aspiration 4:

Our application is highly integrated with Facebook. Our application retrieves information about the user's posts, likes, comments, events and their list of friends. This data is processed and displayed in the form of the various graphs that are seen by the user.

Bonus: Our application uses the Javascript method of visibility control. The reasons why we have chosen this approach is that using the Javascript SDK we can transition between pages without having to reload the entire page which would make the application considerably faster. This approach also helps us to ensure that the url viewed is neat. The PHP SDK should be used whenever one wants to extract information from Facebook and make calls to the graph api.

Aspiration 5:

The database schema that we used for our application is as follows –

```
CREATE TABLE types (
        tid INTEGER NOT NULL AUTO_INCREMENT,
```
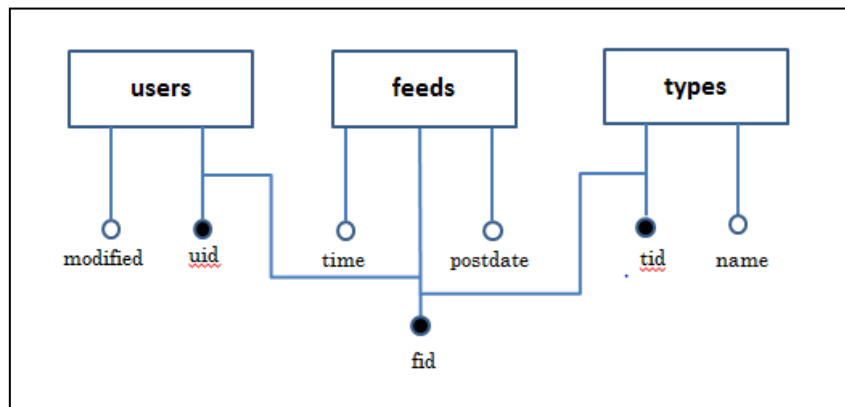
```
        name VARCHAR(20) NOT NULL,
        PRIMARY KEY(tid)
);

CREATE TABLE users(
        uid VARCHAR(100) NOT NULL,
        modified DATE NOT NULL,
        PRIMARY KEY(uid)
);

CREATE TABLE feeds(
        fid VARCHAR(100),
        uid VARCHAR(100) NOT NULL,
        tid INTEGER NOT NULL,
        postdate DATE,
        time TIME,
        PRIMARY KEY(fid),
        FOREIGN KEY(tid) REFERENCES types(tid),
        FOREIGN KEY(uid) REFERENCES users(uid)
);
```


Bonus: Primary key of *home_faculties* table is a combination of the *matric_no* and *faculty*

Aspiration 6:

A few of the SQL queries that we used in our application are as follows –

```
"SELECT tid, count(*)

     FROM feeds
     WHERE uid = ".$uid."
     GROUP BY tid;"
```

This query is used to categorize the number of posts made by a user on the basis of the types of each post. This type of a post could be 'status', 'photo', 'link', 'video' or 'swf' which are retrieved from the types table.

```
"SELECT postdate, count(*)
                FROM feeds
                WHERE postdate BETWEEN ('".$firstDateLastMonth." 00:00:00') AND
   ('".$dateToday." 23:59:59')
                AND uid = ".$uid."
                GROUP BY postdate;"
```

This query is used to retrieve information regarding the number of posts by a particular user grouped by each date in the last two months.

```
"SELECT count(*)
        FROM feeds
        WHERE postdate BETWEEN ('".$firstDateLastMonth."') AND ('".$dateToday."')
                AND uid = ".$uid."
AND time BETWEEN ('".$timeSlotLimits[0]."') AND ('".$timeSlotLimits[1]."');""SELECT
   postdate, count(*)
                FROM feeds
                WHERE postdate BETWEEN ('".$firstDateLastMonth." 00:00:00') AND
   ('".$dateToday." 23:59:59')
                AND uid = ".$uid."
                    GROUP BY postdate;"
```

This query is used to get the number of posts that have been made by a particular user between a certain time slot. This data is used to build the pie-chart that is seen under the activity level tab of our application.

Aspiration 7:

The graph queries that we have used in our application include the following –

```
$request = new FacebookRequest(
                            $session,
                            'GET',
        '/'.$uid.'/posts?fields=id,created_time,likes.limit(1).summary(true),type&
        since='.$starttime.'&until='.$endtime.'&limit='.$limit);
```

This is by far the most interesting graph query that we have used. This query searches through the user's posts and returns the details of those posts that have been last updated between the timeframe provided by $starttime and $endtime which are both in the unix timestamp format. This query returns a summary of the number of likes and comments of each post along with the id and the created_time data of the post. The segment .limit(1) for likes and comments restricts the number of users returned who have liked/commented on the post. This helps to speeden the query time as the amount of data received is significantly reduced.

```
$request = new FacebookRequest(
                        $session,
                        'GET',
```

```
        "/".$uid."/events?fileds=id,start_time,end_time&since=$starttime&unt
il=$endtime");
```

This graph query retrieves the details of the events that the user has accepted invitations to between the specified timeframe.

Aspiration 8:

From our application, we allow the user to share their graphs generated and post them on their Facebook feed.

Aspiration 9:

Our application contains the like button which has been placed at the bottom of the right hand side bar. This location of the like button is the most optimal due to the static nature of this bar. The like button is thus always visible on the screen even when the user scroll's down while viewing the ranking of his latest posts.

Aspiration 10:

Our application has a feature implemented that allows the user to invite their friends to the application. On clicking the invite friend button, a pop-up appears which enables you to invite friends to the application. Moreover, user can also select the types of their friends including "Suggested Friends", "All Friends", "Percenata Users" and "Friends to Invite" in the pop-up.

Apart from this feature, we also have the view friends button in our subpages which allows the user to view all his friends who are users of our application. In addition, after the user merely clicks on one of the friends' avatar, he/she can view their friends' graphs for his/her current topic. However, it is not possible to view friends that aren't users of this application due to Facebook's latest restrictions on retrieving information of friends in v2.0.

Aspiration 11:

Since we do not store user sensitive data like post message, photo etc in our database, we didn't provide an option for user to delete their accounts. Inside our user table, only user id and last updated time are kept, so we totally protect user data and post message are dynamically retrieved when needed.

Aspiration 12:

We embedded Google Analytics in our login.html and index.php, since other pages are all subpages of index.php.
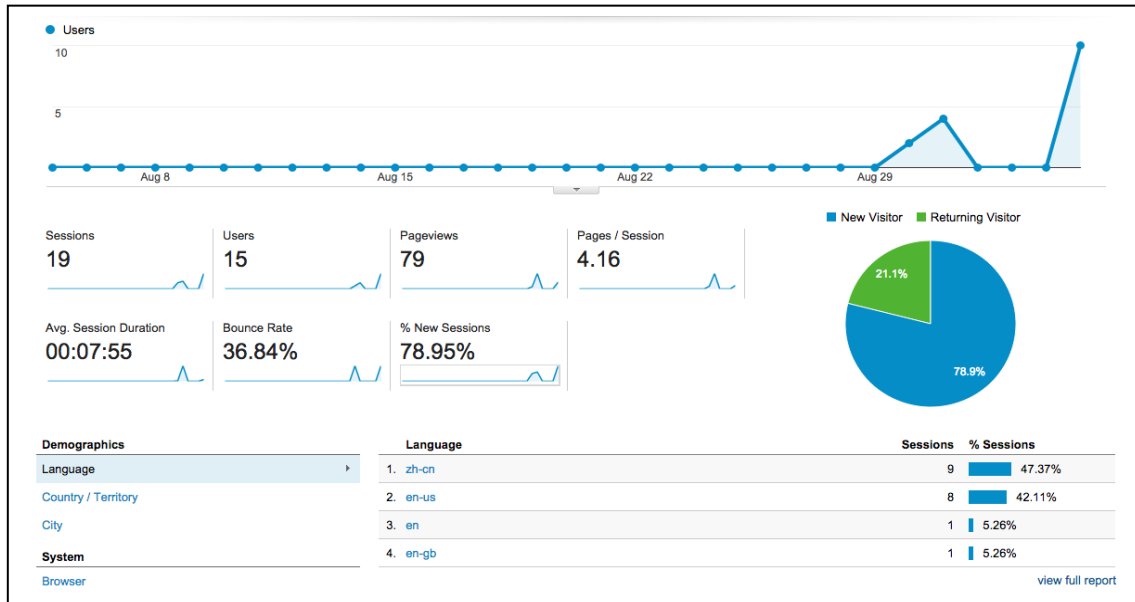
Country / Territory	Sessions	% Sessions
1. 🇸🇬 Singapore	19	100.00%

view full report

Operating System	Sessions	% Sessions
1. Macintosh	10	52.63%
2. Windows	5	26.32%
3. iOS	3	15.79%
4. Android	1	5.26%

view full report

Browser	Sessions	% Sessions
1. Safari	9	47.37%
2. Chrome	7	36.84%
3. Internet Explorer	2	10.53%
4. Android Browser	1	5.26%

view

Aspiration 13:

We have tried our best to ensure the best user interaction experience. At every step on the development, we have tried to keep our final goal aligned to the comfort of the end user. Here is a link to a video that demonstrates the user interaction experience of our application. This video covers several user interactions, including login, viewing the graphs, viewing graphs pertaining to the user's friends using the application, reading the user's most popular posts and also viewing it directly on Facebook

http://www.youtube.com/watch?v=1OkAO39WcS8&feature=youtu.be

Aspiration 14:

All of page transitions are implemented by JQuery DOM manipulation. Controller switches views according to user actions, and this greatly improves user experience by reducing loading time significantly, because reloading the whole page will request all resources again, like background etc.

For the Most Popular Posts page, we are dynamically adding and removing posts content to list elements.

```
function renderRanking(name, type, index, like, data){
      var time = data.created_time;
      var link = data.actions[0].link;
      var message = '';
      /* display ranking-data */
      $('#'+index).show();
      if (data.message) {
            message = data.message;
            $('#'+index+' .message').html(message);
      } else {
            $('#'+index+' .message').html(message);
      }
      $('#ranking-title').html(type+ ' - '+name);
      $('#like-ranking-'+index).html(like+' likes');
      $("#read-more-"+index).attr("href", link);
}
```

Initially every list element is hidden because for those users who don't have 8 posts, some of the empty list elements should not be shown, hence only after we have received the content for a post, we make a list element as visible. For each post, we dynamically insert its message, number of likes and link to the corresponding containers.

Aspiration 15:

For every graph generated, users can share it to their feeds and appear in their timeline. These shares generate clicks and traffic towards the site. These shares allow users to tell stories about their Facebook usage where the objects in question involve the interaction between the user and the user's friends.